

12 . 13 - 99

A

JC542-U.S. PTO
12/09/99

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship..... Appiah et al.
Applicant..... Microsoft Corporation
Attorney's Docket No. MS1-435US
Title: Printer Driver Identification for a Remote Printer

JC584 U.S. PTO
09/454221
12/09/99

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks,
Washington, D.C. 20231

From: Allan T. Sponseller (Tel. 509-324-9256; Fax 509-323-8979)
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

1. Specification--title page, plus 28 pages, including 34 claims and Abstract
2. Transmittal letter including Certificate of Express Mailing
3. 4 Sheets Formal Drawings (Figs. 1-4)
4. Return Post Card

Large Entity Status ☒

Small Entity Status ☐

Date: 12/9/99

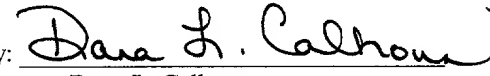
By: 
Allan T. Sponseller
Reg. No. 38,318

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable) EL427262607

Date: 12/9/99

By: 
Dana L. Calhoun

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Printer Driver Identification For A Remote Printer

Inventor(s):
Madan Appiah
Tad Brockway
Ritu Bahl

ATTORNEY'S DOCKET NO. MS1-435US

TECHNICAL FIELD

This invention relates to server-client systems and, in particular, to a server-client system in which a proper printer driver for a printer attached to the client is identified.

BACKGROUND OF THE INVENTION

In a typical server-client system, a server computer is connected to several client computers or terminals via a network. In such a system, applications are executed at the server rather than at the client. The server transmits display information to the client for display at the client, and user inputs (e.g., via keyboard or mouse) at the client are transmitted to the server for processing by an application(s) executing at the server. The "client" may be an entire client computer, or alternatively an application executing at the client computer.

A server in such a system typically has an operating system that can run several client sessions concurrently. Each client user has access to various resources of the server, including the processor, data storage, application programs, etc. Software applications that are resident on the server are available to each client for independent execution by the client. Each session is independent from the other sessions and, therefore, one client cannot access information relating to another client. In this manner, the server provides a logically independent machine for each client connected to the server.

Client users frequently use peripheral devices physically attached to the client machines. For example, a user may wish to attach a printer to the user's client computer (a "local" printer) in order to print data generated by an application that is running on the server computer. To do this, currently the user

1 must manually install the local printer and redirect the printer queue created by the
2 server to the I/O port of the client computer to which the printer is connected.
3 Such manual installation of peripheral devices is undesirable because it requires
4 significant time and effort on the part of the user. A co-pending application (U.S.
5 Patent Application No. 09/_____) entitled "Automatic Detection And Installation
6 Of Client Peripheral Devices By A Server", to Tad Brockway, Madan Appiah,
7 Adam Overton, and Ritu Bahl, filed concurrently herewith, (Attorney Docket No.
8 MS1-432US) describes a system which resolves many of these manual installation
9 problems by automatically detecting such devices and installing and
10 corresponding device drivers at the server.

11 However, one problem that can be encountered with either manual
12 identification or automatic detection of peripheral devices is the identification of
13 the proper driver to be installed. In order to properly install a peripheral device,
14 the correct driver needs to be installed at the server. Typically, the client notifies
15 the server of the correct driver to use for a particular device. However, situations
16 can arise where the driver that the client indicates should be used for a particular
17 peripheral device is not available at the server, thereby preventing installation of
18 that particular device.

19 The invention described below addresses these disadvantages, providing for
20 printer driver identification for a remote printer.

21 22 **SUMMARY OF THE INVENTION**

23 In a server-client environment, a peripheral device can be attached to a
24 client computer. The server selects a corresponding printer driver to install on the
25 server to allow applications executing on the server to print to the printer attached

1 to the remote client computer. For purposes of discussion, reference herein is
2 made primarily to printers. However, the driver selection described herein can be
3 analogously applied to any of a variety of peripheral devices, such as scanners,
4 card readers, zip drives, etc.

5 An embodiment of the server selects a printer driver to install at the server
6 for use with the printer attached to the remote client computer using an "exact
7 match" technique. In this embodiment, the server receives a printer driver
8 identifier and checks whether any of a plurality of printer drivers maintained at the
9 server has a corresponding identifier that matches the received printer driver
10 identifier. If such a match is found, then the matching printer driver from the
11 plurality of printer drivers is installed on the server.

12 Another embodiment of the server selects a printer driver to install at the
13 server for use with the printer attached to the remote client computer using a
14 "driver name mapping" technique. In this embodiment, the server receives a
15 printer driver identifier and checks whether any of a plurality of printer drivers
16 maintained at the server has a corresponding identifier that is different than the
17 received printer driver identifier but that corresponds to the same printer driver as
18 the received printer driver identifier. Such differences can occur, for example,
19 when a manufacturer or distributor of a printer driver changes the name of the
20 printer driver. If such a match is found, then the matching printer driver from the
21 plurality of printer drivers is installed on the server.

22 Another embodiment of the server selects a printer driver to install at the
23 server for use with the printer attached to the remote client computer using a
24 "close match" technique. In this embodiment, the server receives a printer driver
25 identifier and checks whether any of a plurality of printer drivers maintained at the

1 server has a corresponding driver name that is the same as a driver name received
2 as part of the printer driver identifier. If such a match is found, then the matching
3 printer driver from the plurality of printer drivers is installed on the server without
4 regard for whether that particular printer driver has a corresponding driver version
5 that is the same as a driver version received as part of the printer driver identifier.

6 Another embodiment of the server issues a notification when a printer
7 driver is installed that is not an exact match to a received printer driver identifier.
8 Such an "inexact" match could be due to a change in driver name, installation of a
9 previous version of a requested printer driver, installation of a more recent version
10 of a requested printer driver, etc. The notification can be used, for example, to
11 make a system administrator or the user of the client computer aware that the
12 inexact match occurred.

13 Another embodiment of the server attempts to automatically update a
14 printer driver under certain circumstances. In this embodiment, if a received
15 printer driver identifier indicates a more recent version of a printer than is
16 available on the server, the server attempts to locate the more recent version and
17 add it to the server so that the most recent version can be installed on the server.

18 19 **BRIEF DESCRIPTION OF THE DRAWINGS**

20 The present invention is illustrated by way of example and not limitation in
21 the figures of the accompanying drawings. The same numbers are used
22 throughout the figures to reference like components and/or features.

23 Fig. 1 is a diagrammatic illustration of a server/client system.

24 Fig. 2 is a block diagram illustrating a server/client system in more detail.
25

1 Fig. 3 is a flow diagram depicting an exemplary process for installing
2 peripheral devices at a server that are attached to a client.

3 Fig. 4 is a flow diagram of an exemplary process for installing an
4 appropriate printer driver in accordance with one embodiment of the invention.

5 6 **DETAILED DESCRIPTION**

7 Fig. 1 shows a server/client computer system 30 having a server 32, a first
8 client 34 and a second client 36 interconnected via a network 38. The server 32
9 and the clients 34, 36 have modems or network cards, which facilitate access
10 (direct or otherwise) to the network 38 so that the server 32 and the clients 34, 36
11 can exchange information over the network 38. The clients 34, 36 each have a
12 desktop 40, 42 interface displayed.

13 The server 32 is a computer. A client 34, 36 may be a computer having
14 Plug and Play capability, a computer that is not Plug and Play compatible, or a
15 terminal, which does not have the processing capability of a computer.
16 Implications of utilizing the different types of clients will be discussed in greater
17 detail below.

18 It is noted that, although the server/client computer system 30 depicted in
19 Fig. 1 has a first client 34 and a second client 36, there may be one to any number
20 of clients connected to the server 32 concurrently. The number of clients that may
21 be connected to a server is limited only by the architecture of a particular server.

22 The server 32 is configured to provide a logically independent machine for
23 each client 34, 36 connected to the network 38. That is, the server 32 establishes a
24 session for each client 34, 36, provides the desktop 40, 42 for each client 34, 36,
25 and makes server resources available to the clients 34, 36. Such resources include,

1 but are not limited to, allocations of processor time, memory, data storage, video
2 processing, application programs, etc. A user of either of the clients 34, 36
3 interacts with the desktop 40, 42 on the client 34, 36 to run software applications
4 that reside on the server 32. While the user provides input to and receives output
5 from the client 34, 36, most processing is performed at the server 32.

6 The network 38 provides a communications link between the server 32 and
7 the clients 34, 36 through which data is transmitted. The network 38 may be a
8 local area network (LAN), a wide area network (WAN), the Internet, or the like,
9 provided that it can accommodate server/client functionality.

10 Fig. 2 shows a server/client system 50 having a server 52 and a client 54.
11 The server 52 includes a processor 56 and memory 58. The server 52 also has a
12 network port 60, which facilitates access to the network 38. The network port 60
13 may be implemented as a modem, network card, or other device, which interfaces
14 the server 52 to the network 38.

15 The server 52 can be implemented as a common personal computer or other
16 type of computer, such as a laptop computer, etc. The server 52 runs an operating
17 system 62, which is stored in memory 58 and executes on the processor 56. The
18 operating system 62 is a multitasking operating system such as a Windows® brand
19 operating system from Microsoft Corporation (e.g., Windows® 98, Windows® 95,
20 Windows® NT, or other derivative of Windows®). However, other multi-tasking
21 operating systems may be used.

22 The server 52 has a printing subsystem 64 implemented in the operating
23 system 62 stored in memory 58. The printing subsystem 64 is used to direct all
24 operations involving printers and printing, including installing printers, creating
25 and managing printer queues, removing printers, uninstalling printers, etc. The

1 printing subsystem includes a peripheral mapping unit 66 that is configured to
2 map, or redirect, printer queues to printer ports. The printing subsystem 64 may
3 be an integral part of the operating system 62 or it may be a software function that
4 is callable by the operating system 62 or by application programs.

5 The server 52 is shown having a printer driver 70 and a printer queue 72
6 installed and resident within the memory 58. It is noted, however, that the server
7 52 will only include the printer driver 70 and the printer queue 72 when a printer
8 has been connected to the system 50 and installed on the server 52. The printer
9 driver 70 is a printer-specific software program that provides an interface between
10 a printer and the server 52 and allows the server 52 to provide print functions via
11 the printer. When the printer driver 70 is installed on the server 52, the printer
12 queue 72 is created. The printer queue 72 accepts print jobs from the server 52
13 and queues them for printing.

14 The server 52 further includes a driver library 68, which is a set of multiple
15 peripheral device drivers that are available for installation on the server 52 (e.g., as
16 the printer driver 70). In the illustrated example, the driver library 68 is
17 maintained at the server 52. Alternatively, the driver library may be maintained at
18 some other location (e.g., a remote location accessible via the network 38) that is
19 accessible to the server 52. When a driver is installed on the server 52 for a device
20 attached to the client 54, the driver is obtained from driver library 68.

21 The operating system 62 further includes a driver matching module 69.
22 When a driver is to be installed on the server 52, the server 52 receives an
23 indication (e.g., automatically detected by the client 54 or manually input by a
24 user) of the driver that is to be installed. However, situations can arise when the
25 received indication identifies a driver that is not available to the server 52 (e.g., is

1 not in the driver library 68) and thus is not available for installation at the server
2 52. The driver matching module 69 remedies such situations by attempting to
3 select, for installation at the server 52, a driver that most closely matches the
4 driver identified in the received indication.

5 The server 52 further includes a driver mapping 71. The driver mapping 71
6 is a record (e.g., a list) of old driver names and new driver names. Situations can
7 arise where the name of a driver is changed. Such changes are maintained by the
8 server 52 in order to install the appropriate driver in the event that the driver is
9 identified by its old name, as discussed in more detail below.

10 The client 54 includes memory 76 and a display 78. Similar to the server
11 52, the client 54 also has a network port 79 to facilitate access to the network 38.
12 The network port 79 may be implemented as a modem, network card, or other
13 device, which interfaces the client 54 to the network 38. The client 54 also
14 includes a printing subsystem 82 resident in the memory 76, a first I/O port 84, a
15 second I/O port 86, and a printer 88 connected to the first I/O port 84. The
16 printing subsystem supports the concept of printing queues, which are logical
17 containers that encompass a physical printer's driver information, properties, and
18 pending print jobs.

19 The client 54 has a system registry 90 located in the memory 76. The
20 printing subsystem 82 may be an integral part of the operating system 80 or it may
21 be a software function that is callable by the operating system 80 or by application
22 programs. The server/client system 50 also includes the printer 88, which is
23 connected to the first I/O port 84. A printer queue 92 is installed in the memory
24 76 for the printer 88.
25

1 It is noted that, although the client 54 is depicted as being a computer, the
2 client 54 may also be a terminal, which does not have the mass storage or
3 processing capabilities of a computer. The present invention functions similarly
4 whether the client 54 is a computer or a terminal.

5 The client 54 may automatically detect the printer 88, or the printer 88 may
6 be manually identified to the client 54 and/or the server 52. For example, a user
7 may manually notify the server 52 or the client 54 of the location of the printer 88
8 and an I/O port of the client 54 to which the printer 88 is attached (e.g., by use of a
9 printer setup "wizard").

10 When the printer 88 is attached to the client 54, information identifying a
11 corresponding printer driver to use for the printer 88 is provided to the server 52.
12 This information can be automatically detected by the client 54 or alternatively
13 can be supplied manually by the user of the client 54. Additionally, the
14 information can be provided when the user of the client 54 logs on to the server
15 52, or alternatively at some subsequent time. Regardless of the source of the
16 printer driver information or when the information is provided, this received
17 printer driver information is used by the server 52 in selecting an appropriate
18 printer driver from the driver library 68 to install on the server 52 in order to allow
19 applications executing on the server 52 to print to the printer 88.

20 Fig. 3 is a flow diagram depicting an exemplary process for installing
21 peripheral devices at a server that are attached to a client. The method is described
22 with continuing reference to Figs. 1 and 2.

23 At step 200 in Fig. 3, the client 54 establishes a connection with the server
24 52. After the client 54 establishes the connection, the client 54 waits for an
25 announcement package from the server 52. When the server 52 detects the

1 connection from the client 54 (step 202), the server 52 sets up a session for the
 2 client 54 at step 204. At step 206, the server 52 initializes the data structures
 3 necessary to provide the session for the client 54. When this is completed, the
 4 server 52 sends an announcement package to the client 54, which signals the client
 5 54 that it can commence interactive communications with the server 52.

6 At step 208, the printer(s) attached to the client 54 are identified. As
 7 previously discussed, the client 54 is a computer having a printing subsystem. The
 8 printing subsystem provides an interface, such as an Application Programming
 9 Interface (API) that allows applications (such as operating system 80 or another
 10 application separate from operating system 80) to identify printer queues installed
 11 at the client 54 (e.g., the printer 88). Such queues can be installed at client 54 in
 12 any of a variety of manners, such as automatically (e.g., using a Plug and Play
 13 (PnP) subsystem), or manually. At step 210, information describing these
 14 identified queues, such as a particular driver name and version or a PnP identifier,
 15 is sent to the server 52.

16 At step 214, the server 52 installs the identified printer(s). The printing
 17 subsystem 64 on the server 52 parses the information to determine the printer
 18 driver corresponding to the printer 88 that should be installed on the server 52.
 19 This identified printer driver is then used to determine which printer driver from
 20 library 68 to install as printer driver 70, as discussed in more detail below. The
 21 printer driver 70 is installed on the server 52 and the printer queue 72 is created.
 22 The peripheral mapping unit 66 then redirects the printer queue 72 to the I/O port
 23 84 to which the printer 88 is connected.
 24
 25

1 Fig. 4 is a flow diagram illustrating an exemplary process for installing an
2 appropriate printer driver in accordance with one embodiment of the invention.
3 Fig. 4 is described with additional reference to components in Fig. 2.

4 Initially, the server 52 obtains the printer driver information (step 422),
5 which identifies a printer driver that is expected to be installed on the server 52 for
6 use with the printer 88. This “expected driver” can be determined in various
7 manners. For example, the user of the client 54 may indicate a particular printer
8 driver to use, or alternatively a PnP subsystem on the client 54 may indicate that a
9 particular printer driver is to be used (e.g., by driver name and version number, by
10 a PnP identifier that can be resolved by server 52 into a driver name and version
11 (e.g., via a lookup table), etc.). The printer driver information includes both a
12 printer driver name and a printer driver version. Typically the printer driver
13 information for one printer will identify a single printer driver, although
14 alternatively multiple printer drivers (including multiple names and multiple
15 versions) may be identified in the printer driver information.

16 Situations can arise where the identified printer driver is not available on
17 the server 52. Such situations can arise for a variety of different reasons. For
18 example, the name of the printer driver (and/or the version of the printer driver)
19 may have been changed by the manufacturer (or developer, distributor, etc.) of the
20 driver, so that the identified printer driver name no longer exists. By way of
21 another example, the version of the identified printer driver may be a new or older
22 version of the printer driver than is available on the server 52.

23 Thus, the server 52 attempts to match the received printer driver
24 information to a particular printer driver available to the server 52 (step 424). The
25 server 52 then takes different actions depending on whether the attempt to match

1 the received printer driver information to a particular printer driver is successful
2 (step 426). If the attempt to match is successful, then the matching printer driver
3 is installed at the server (step 428). However, if the attempt to match is
4 unsuccessful, then the driver installation process fails (step 430).

5 The server 52 uses various techniques to attempt to match the received
6 printer driver information to a particular printer driver. In the illustrated example,
7 three such techniques are used, referred to as "exact match" (step 432), "driver
8 name mapping" (step 434), and "close match" (step 436). Each of these
9 techniques involves comparing at least a portion of the received printer driver
10 information to information corresponding to the printer drivers available to the
11 server 52. Such comparisons can be done by accessing each of the printer drivers
12 individually, by accessing a record of the printer driver information for each of the
13 available printer drivers (having identifiers or links to the corresponding printer
14 driver(s)), or by a combination thereof.

15 The server 52 may follow a particular ordering in using these techniques,
16 such as using the exact match technique first, followed by the driver name
17 mapping and close match techniques if necessary. Alternatively, two or more of
18 these techniques could be used concurrently.

19 In the exact match technique (step 432), the driver matching module 69
20 searches for an exact match between the received printer driver information and
21 the information corresponding to the printer drivers available to the server 52. An
22 exact match between the received printer driver information and a particular
23 printer driver available to the server 52 refers to the printer driver names being the
24 same and the printer driver versions being the same.

1 In the driver name mapping technique (step 434), the driver matching
2 module 69 searches for a change in driver name. Situations can arise where the
3 name of a particular driver is changed. The server 52 maintains a record (e.g.,
4 driver mapping 71) of changes to driver names. This record includes the previous
5 name(s) for printer drivers as well as their new names. If a received driver name
6 is included in this record as a previous name, then the server 52 searches the
7 available printer drivers for the new name identified from this record.

8 For example, assume that at the time the printer 88 is initially attached to
9 the client 54 the correct name for the printer driver for the printer 88 is "driverY",
10 and this information is recorded at the client 54. Subsequently, the manufacturer
11 of the printer 88 changes its naming convention, so that the correct name for the
12 printer driver for the printer 88 is changed to "X printer driver". Further assume
13 that the server 52 is distributed after the naming convention change occurs, so the
14 server 52 has available to it the printer driver with the name "X printer driver" but
15 does not have "driverY" available to it. In this situation, a record of the change in
16 printer driver name is maintained by the server 52 (e.g., the manufacturer making
17 the change in naming convention notifies a system administrator of the change in
18 naming convention or the old and new driver names for the printer drivers affected
19 by the naming convention change). Thus, if the client 54 subsequently identifies
20 to the server 52 "driverY" in its printer driver information, the server 52 knows
21 that "X printer driver" is the matching printer driver to be installed.

22 The driver name mapping technique can be used as an additional technique
23 in the event the exact match technique (and/or the close match technique) fails.
24 Alternatively, the driver name mapping technique may be used prior to either the
25 exact match technique or the close match technique, with the new driver name

1 resulting from application of the driver name mapping technique being used as the
2 received driver name (and possible driver version) during application of the exact
3 match and close match techniques.

4 In the close match technique (step 436), the driver matching module 69
5 searches for an available printer driver with the same name as the received printer
6 driver but a different driver version. Situations can arise where one of the server
7 52 and the client 54 is aware of a more recent version of the printer driver than the
8 other is aware of. In such situations, as long as the printer driver name matches,
9 the server 52 will select the version of the printer driver it has available to it for
10 installation. This selection by the server 52 occurs regardless of whether the
11 version identified by the client 54 is a newer version or an older version than the
12 version available to the server 52.

13 For example, the client 54 may be aware of version 2.1 of “driverZ” and
14 identify to the server 52, as the printer driver information for a particular printer,
15 version 2.1 of “driverZ”. However, the server 52 may only have version 2.0 of
16 “driverZ” and not yet have version 2.1 available to it. In this situation, the server
17 52 selects version 2.0 of “driverZ” for installation. By way of another example,
18 the client 54 may be aware of version 3.10 of “driverA” and identify to the server
19 52, as the printer driver information for a particular printer, version 3.10 of
20 “driverA”. However, the server 52 may have a newer version of “driverA”
21 available to it, version 3.11. In this situation, the server 52 selects version 3.11 of
22 “driverA” for installation.

23 Alternatively, the driver name mapping technique (step 434) and the close
24 match technique (step 436) can be combined into a single step. The record
25 maintained by server 52 of changes to driver names (e.g., driver mapping 71) can

1 also include mappings of "requested driver to available driver" for all clients or
2 alternatively on a per-client basis. For example, server 52 may maintain a record
3 that when a request is made for version 3.10 of "driverA", that the driver to be
4 installed is version 3.11 of "driverA". Such mapping information can be identified
5 for storage at server 52 in a variety of manners, such as manual identification by a
6 system administrator, automatic detection the first time it arises (e.g., close match
7 technique step 436), etc.

8 The server 52 may also optionally take additional actions depending on
9 which of the various mapping techniques were used in obtaining a printer driver
10 match. One such action is to issue a notification in the event that anything other
11 than the exact match technique was used in selecting the printer driver for
12 installation. The notification could simply be a warning that a technique other
13 than exact match was used, or alternatively may include additional information
14 describing the matching process or results, such as what technique(s) was used,
15 what driver was selected as the matching driver, what printer driver information
16 was received, etc. Such notifications can be made in any of a wide variety of
17 forms, such as: sending an electronic mail message to the user of the client 54 or a
18 system administrator; displaying a dialog box on either the client 54 or the server
19 52; sounding an audible "alarm" at the server 52; adding an entry to a notification
20 log; placing a telephone call (e.g., to a phone or pager); etc.

21 Another action that may be taken is to automatically attempt to add a new
22 driver to the drivers available to the server 52. For example, if the received printer
23 driver information identifies a newer version of a particular driver that is not
24 available to the server 52, the server 52 can automatically attempt to obtain that
25 new driver. The new driver can be obtained in any of a variety of manners, such

1 as accessing a remote server computer for the distributor (or manufacturer, etc.) of
2 the printer driver via the Internet, via a direct-dial connection, etc. The server 52
3 can be programmed with information identifying how to connect to such a remote
4 server (e.g., including a telephone number, uniform resource locator (URL), etc.).
5 Once the new driver is obtained, it could be installed at the server 52 immediately,
6 or alternatively could be used the next time the client 54 logs on to the server 52
7 (so that subsequent matching attempts would result in an exact match).

8 9 **Conclusion**

10 In a server-client environment as described herein a printer can be attached
11 to a client computer and made available for printing to by applications executing
12 on the server computer. The server receives information identifying a printer
13 driver to be used for the printer and attempts to identify a closest matching printer
14 driver to be installed at the server.

15 Although the invention has been described in language specific to structural
16 features and/or methodological steps, it is to be understood that the invention
17 defined in the appended claims is not necessarily limited to the specific features or
18 steps described. Rather, the specific features and steps are disclosed as preferred
19 forms of implementing the claimed invention.

1
2 **CLAIMS**

3 1. A method in a server-client environment, the method comprising:
4 receiving a driver identifier for a printer attached to the client;
5 using the driver identifier to select a closest matching driver of a plurality
6 of drivers to install at the server;
7 installing, at the server, the selected driver; and
8 allowing applications executing on the server to print to the printer using
9 the installed driver.

10
11 2. A method as recited in claim 1, wherein the receiving comprises
12 receiving the driver identifier from the client.

13
14 3. A method as recited in claim 1, wherein the driver identifier includes
15 both a driver name and a driver version.

16
17 4. A method as recited in claim 1, wherein the using comprises
18 accessing a library at the server that stores the plurality of drivers.

19
20 5. A method as recited in claim 1, wherein:
21 the using comprises checking whether any of the plurality of drivers has a
22 corresponding driver identifier that is the same as the received driver identifier;
23 and
24
25

1 if a particular driver of the plurality of drivers has a corresponding driver
2 identifier that is the same as the received driver identifier, then selecting that
3 driver to install at the server.

4
5 **6.** A method as recited in claim 1, wherein:

6 the using comprises checking whether any of the plurality of drivers
7 currently has a corresponding driver identifier that is different than the received
8 driver identifier but that corresponds to the same driver as the received driver
9 identifier; and

10 if a particular driver of the plurality of drivers currently has a corresponding
11 driver identifier that is different than the received driver identifier but that
12 corresponds to the same driver as the received driver identifier, then selecting that
13 driver to install at the server.

14
15 **7.** A method as recited in claim 6, wherein one of the plurality of drivers
16 currently has a corresponding driver identifier that is different than the received
17 driver identifier but that corresponds to the same driver because of a driver name
18 change by a source of the driver.

19
20 **8.** A method as recited in claim 6, further comprising:

21 issuing a notification that the selected driver currently has a corresponding
22 driver identifier that is different than the received driver identifier but that
23 corresponds to the same driver as the received driver identifier.

1 **9.** A method as recited in claim 1, wherein:

2 the receiving comprises receiving a driver name and a driver version;

3 the using comprises checking whether any of the plurality of drivers has a
4 corresponding driver name that is the same as the received driver name; and

5 if a particular driver of the plurality of drivers has a corresponding driver
6 name that is the same as the received driver name, then selecting that driver to
7 install at the server.

8
9 **10.** A method as recited in claim 9, further comprising:

10 selecting a first driver with a corresponding driver name that is the same as
11 the received driver name to install at the server without regard for whether the
12 received driver version is the same as a corresponding driver version of the first
13 driver.

14
15 **11.** A method as recited in claim 9, further comprising:

16 issuing a notification that the selected driver has a corresponding driver
17 name that is the same as the received driver name but a corresponding driver
18 version that is different than the received driver version.

19
20 **12.** A method as recited in claim 9, further comprising:

21 checking whether the selected driver has a corresponding driver version
22 that is the same as the received driver version; and

23 if the selected driver does not have a corresponding driver version that is
24 the same as the received driver version, then obtaining a new copy of the driver
25 that has the same driver version as the received driver version.

1
2 **13.** A method as recited in claim 12, further comprising obtaining a new
3 copy of the driver only if the received driver version indicates a more recent
4 version of the driver than is indicated by the driver version corresponding to the
5 selected driver.

6
7 **14.** At least one computer-readable memory containing a computer
8 program that is executable by a processor to perform the method recited in claim
9 1.

10
11 **15.** A method implemented in a server in a server-client environment,
12 the method comprising:

13 automatically selecting at least one of a plurality of drivers corresponding
14 to a peripheral device attached to the client; and
15 installing, at the server, the selected at least one driver.

16
17 **16.** A method as recited in claim 15, wherein the peripheral device
18 comprises a printer.

19
20 **17.** A method as recited in claim 15, wherein the automatically selecting
21 comprises using a received driver identifier corresponding to a printer to select a
22 closest matching driver of the plurality of drivers to install at the server.

1 **18.** A method as recited in claim 15, wherein:

2 the automatically selecting comprises checking whether any of the plurality
3 of drivers has a corresponding driver identifier that is the same as a received driver
4 identifier; and

5 if a particular driver of the plurality of drivers has a corresponding driver
6 identifier that is the same as the received driver identifier, then installing that
7 driver at the server.

8
9 **19.** A method as recited in claim 15, wherein:

10 the automatically selecting comprises checking whether any of the plurality
11 of drivers currently has a corresponding driver identifier that is different than a
12 received driver identifier but that corresponds to the same driver as the received
13 driver identifier; and

14 if a particular driver of the plurality of drivers currently has a corresponding
15 driver identifier that is different than the received driver identifier but that
16 corresponds to the same driver as the received driver identifier, then installing that
17 driver at the server.

18
19 **20.** A method as recited in claim 19, further comprising:

20 issuing a notification that the installed driver currently has a corresponding
21 driver identifier that is different than the received driver identifier but that
22 corresponds to the same driver as the received driver identifier.

1 **21.** A method as recited in claim 15, wherein:

2 the automatically selecting comprises checking whether any of the plurality
3 of drivers has a corresponding driver name that is the same as a received driver
4 name; and

5 if a particular driver of the plurality of drivers has a corresponding driver
6 name that is the same as the received driver name, then installing that driver at the
7 server.

8
9 **22.** A method as recited in claim 21, further comprising:

10 selecting a first driver with a corresponding driver name that is the same as
11 the received driver name to install at the server without regard for whether a
12 received driver version is the same as a corresponding driver version of the first
13 driver.

14
15 **23.** A method as recited in claim 21, further comprising:

16 issuing a notification that the installed driver has a corresponding driver
17 name that is the same as the received driver name but a corresponding driver
18 version that is different than the received driver version.

19
20 **24.** A method as recited in claim 21, further comprising:

21 checking whether the installed driver has a corresponding driver version
22 that is the same as a received driver version; and

23 if the selected driver does not have a corresponding driver version that is
24 the same as the received driver version, then obtaining a new copy of the driver
25 that has the same driver version as the received driver version.

1
2 **25.** At least one computer-readable memory containing a computer
3 program that is executable by a processor to perform the method recited in claim
4 15.

5
6 ~~**26.**~~ One or more computer-readable media having stored thereon a
7 computer program that, when executed by one or more processors of a server,
8 causes the one or more processors to:

9 receive a printer driver identifier for a printer attached to a client;

10 use the printer driver identifier to select one of a plurality of printer drivers
11 to install at the server according to the following,

12 if a particular printer driver of the plurality of printer drivers has a
13 corresponding printer driver identifier that is the same as the received
14 printer driver identifier, then selecting that particular driver,

15 if a particular printer driver of the plurality of printer drivers
16 currently has a corresponding printer driver identifier that is different than
17 the received printer driver identifier but that corresponds to the same printer
18 driver as the received printer driver identifier, then selecting that particular
19 printer driver, and

20 if a particular printer driver of the plurality of printer drivers has a
21 corresponding driver name that is the same as a driver name received as
22 part of the printer driver identifier, then selecting that particular printer
23 driver without regard for whether that particular printer driver has a
24 corresponding driver version that is the same as a driver version received as
25 part of the printer driver identifier; and

1 install, at the server, the selected printer driver.

2
3 27. A method as recited in claim 26, wherein the server comprises a
4 terminal server and wherein the client comprises a terminal server client.

5
6 28. A method as recited in claim 26, wherein one of the plurality of
7 printer drivers currently has a corresponding printer driver identifier that is
8 different than the received printer driver identifier but that corresponds to the same
9 printer driver due to a name of the printer driver being changed.

10
11 29. An apparatus comprising:
12 a driver library including a plurality of printer drivers; and
13 a driver matching module to select at least one of the plurality of printer
14 drivers for installation on the apparatus, the selected at least one printer driver
15 corresponding to a printer attached to a client computer.

16
17 30. An apparatus as recited in claim 29, wherein the driver matching
18 module is further to:

19 check whether any of the plurality of drivers has a corresponding driver
20 identifier that is the same as a received driver identifier; and

21 if a particular driver of the plurality of drivers has a corresponding driver
22 identifier that is the same as the received driver identifier, then install that driver
23 on the apparatus.

1 **31.** An apparatus as recited in claim 29, further comprising:
2 a mapping table to map previous driver identifiers to subsequent driver
3 identifiers;

4 wherein the driver matching module is further to check the mapping table
5 to determine whether any of the plurality of drivers currently has a corresponding
6 driver identifier that is different than a received driver identifier but that is a
7 subsequent driver identifier mapped to the received driver identifier as a previous
8 driver identifier; and

9 if a particular driver of the plurality of drivers currently has a corresponding
10 driver identifier that is different than a received driver identifier but that is a
11 subsequent driver identifier mapped to the received driver identifier as a previous
12 driver identifier, then the driver matching module is further to install that driver at
13 the server.

14
15 **32.** An apparatus as recited in claim 29, wherein the driver matching
16 module is further to:

17 check whether any of the plurality of printer drivers has a corresponding
18 driver name that is the same as a received driver name; and

19 if a particular printer driver of the plurality of printer drivers has a
20 corresponding driver name that is the same as the received driver name, then
21 install that printer driver on the apparatus without regard for whether that
22 particular printer driver has a corresponding driver version that is the same as a
23 received driver version.

33. A system comprising:

a client computer having a local printer attached thereto; and

a server computer coupled to the client computer via a network, wherein the server computer includes,

a driver library including a plurality of printer drivers, and

a driver matching module to select at least one of the plurality of printer drivers for installation on the server computer to allow applications executing on the server computer to print to the local printer, the driver matching module selecting one of the plurality of printer drivers for installation based on a printer driver identifier and according to the following,

if a particular printer driver of the plurality of printer drivers has a corresponding printer driver identifier that is the same as the received printer driver identifier, then selecting that particular driver for installation,

if a particular printer driver of the plurality of printer drivers currently has a corresponding printer driver identifier that is different than the received printer driver identifier but that corresponds to the same printer driver as the received printer driver identifier, then selecting that particular printer driver for installation, and

if a particular printer driver of the plurality of printer drivers has a corresponding driver name that is the same as a driver name received as part of the printer driver identifier, then selecting that particular printer driver without regard for whether that particular printer driver has a corresponding driver version that is the same as a

driver version received as part of the printer driver identifier for
installation.

34. A system as recited in claim 33, wherein the client computer is to
transmit the printer driver identifier to the server computer.

1 **ABSTRACT**

2 In a multi-user server-client environment, a printer can be attached to a
3 client computer. The server selects a corresponding printer driver to install on the
4 server in order to allow applications executing on the server to print to the printer
5 attached to the remote client computer. The selection process allows the server to
6 intelligently identify closely matching printer drivers in the event that a particular
7 printer driver that the server is notified of is not available for installation.

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

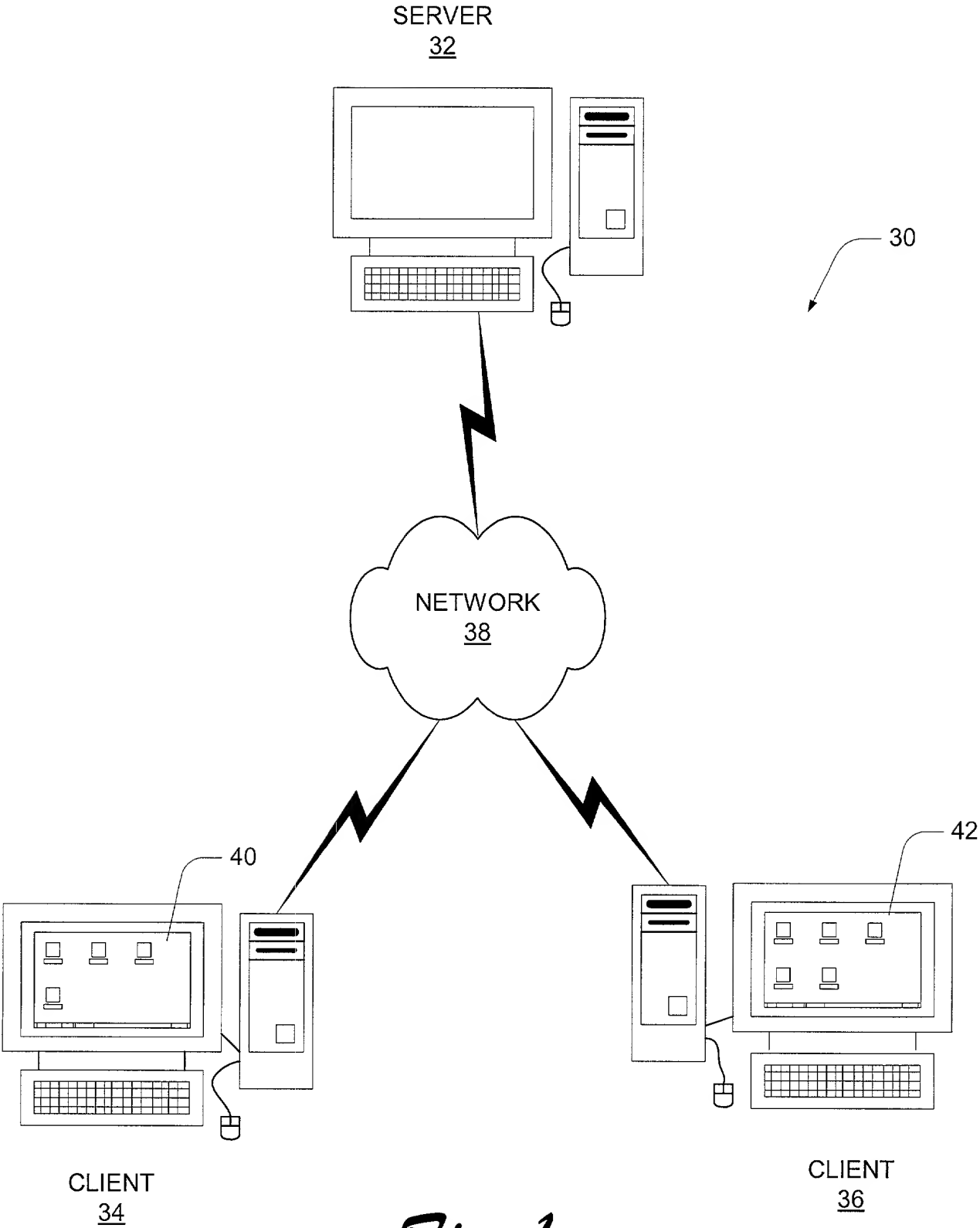
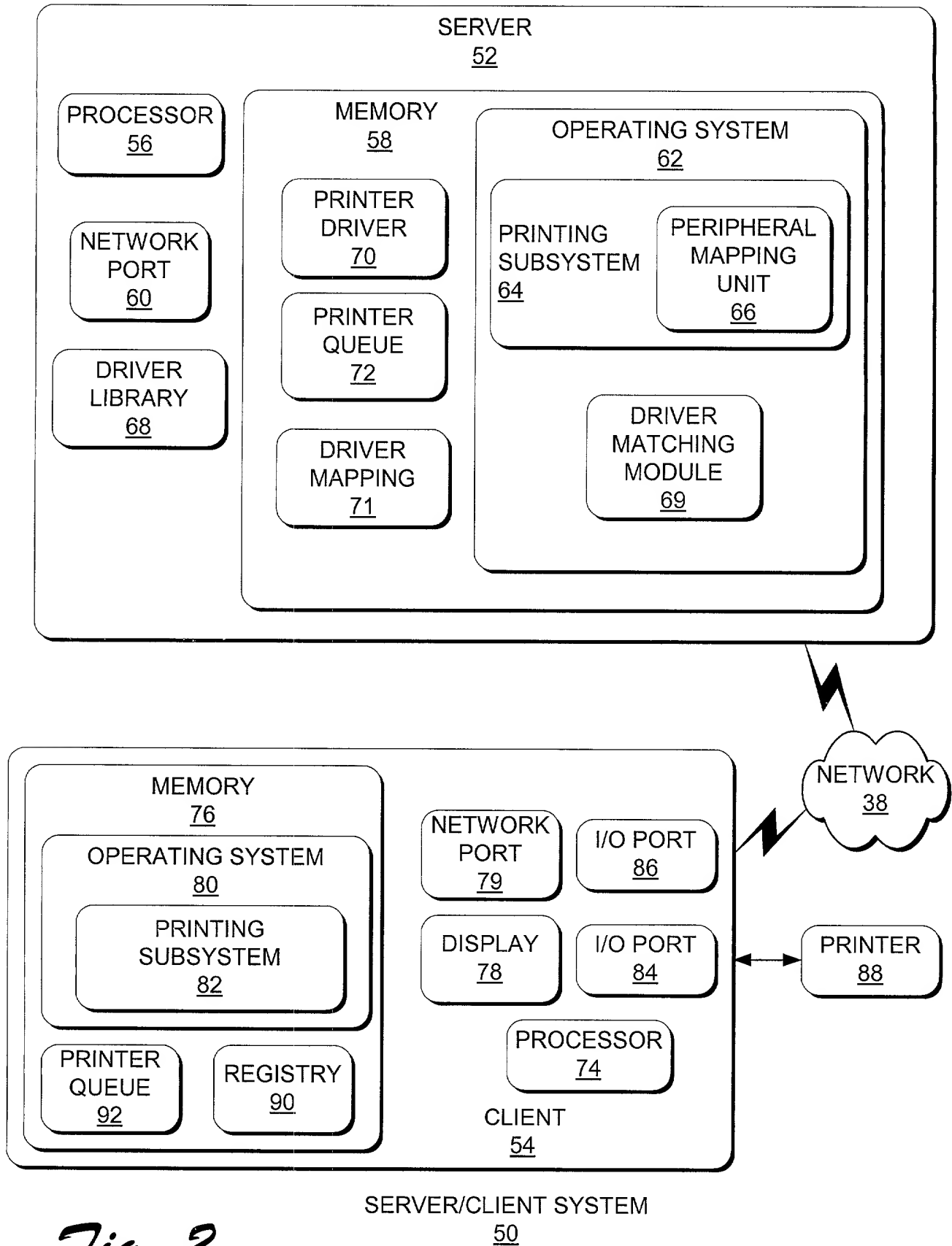
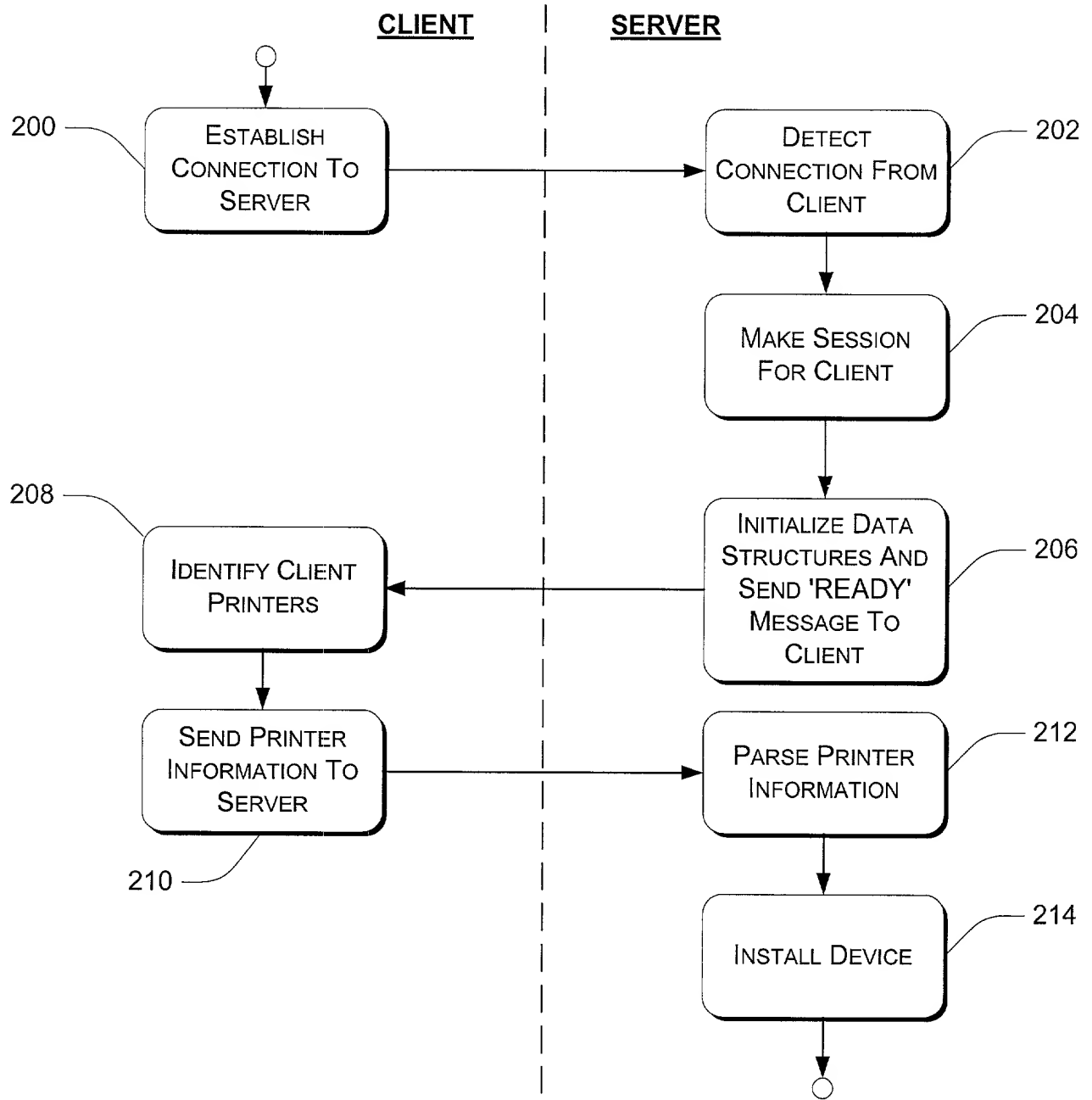
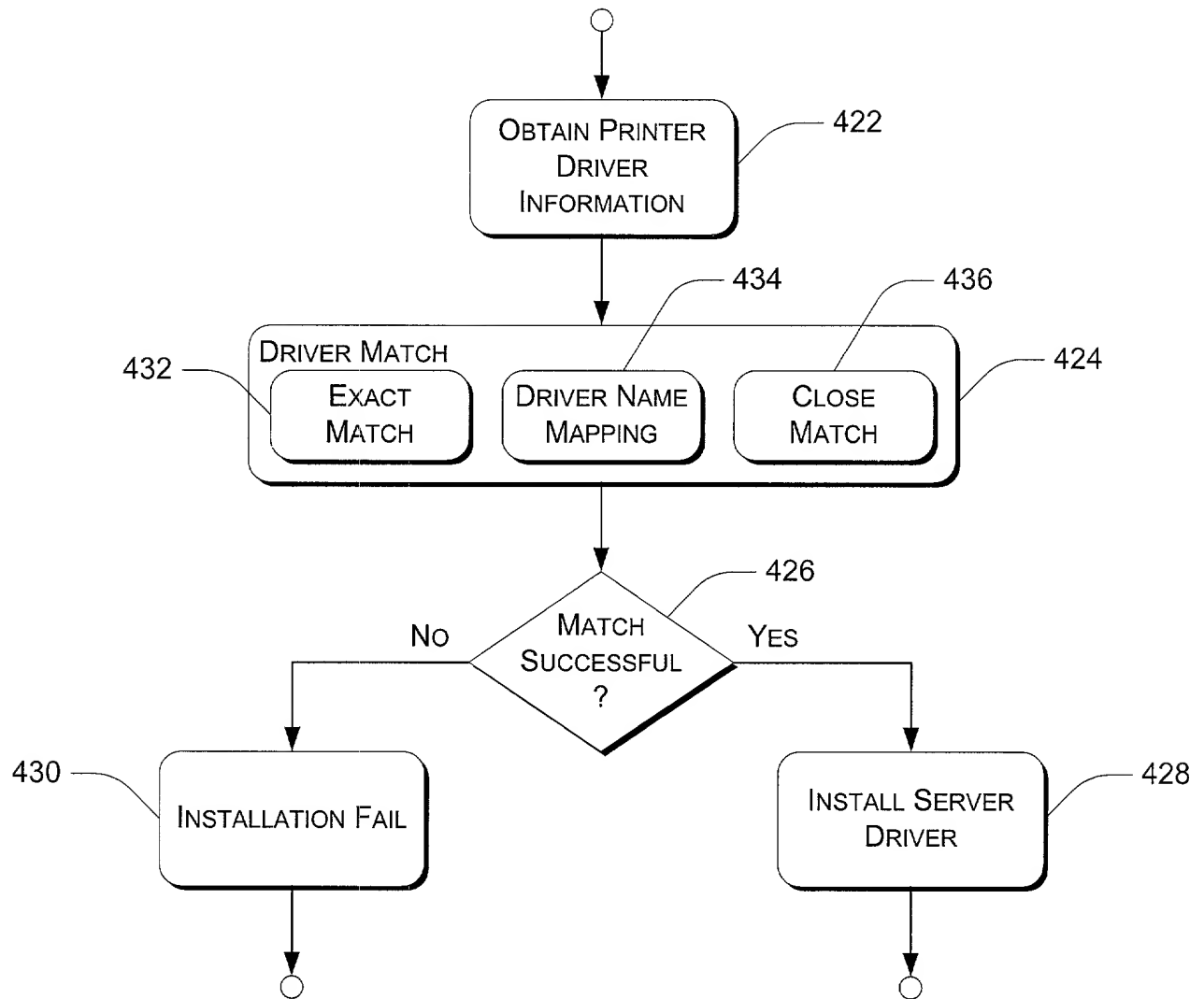


Fig. 1

*Fig. 2*

*Fig. 3*

*Fig. 4*